

CLAIMS

1. A method of configuring a server to provide at least one composite user interface to at least one source application, the composite user interface comprising a plurality of user interface elements provided by said at least one source application, the method comprising, processing a model representing said composite user interface to generate rules for communication between said composite user interface and said at least one source application.
2. A method according to claim 1, wherein said model comprises a model of at least part of a user interface provided by the or each source application and a model of relationships between the at least part of the user interface provided the or each source application and the composite user interface.
3. A method according to claim 1 or 2, further comprising:
storing said rules within a hierarchical data structure comprising a plurality of entities.
4. A method according to claim 3, further comprising:
storing within said hierarchical data structure an entity representing the composite user interface; and
associating with said entity a data group providing configuration data for the composite user interface.
5. A method according to claim 4, further comprising:
storing within said hierarchical data structure a plurality of service entities representing processing modules which are together adapted to process user requests input to said composite user interface to produce one or more requests to at least one source application.
6. A method according to claim 5, wherein at least some of said service entities have an associated data group storing configuration data.

7. A method according to claim 6, wherein one of said service entities is an aggregation service entity representing an aggregation service configured to generate source application requests from a user request.
8. A method according to claim 7, wherein said aggregation service entity comprises:
 - a child entity representing the composite user interface; and
 - said child entity has at least one child entity representing a source application.
9. A method according to any one of claims 3 to 8, wherein said rules are generated using a plurality of writers each writer being associated with an entity in said hierarchical data structure, and being adapted to write data to a data group associated with the respective entity.
10. A method according to claim 9, wherein processing said model comprises:
 - selecting one or more objects within said model;
 - determining one or more writers to be invoked to write data from the or each object to said hierarchical data structure; and
 - invoking the or each writer to write data to said hierarchical data structure.
11. A method according to claim 10, further comprising:
 - determining from said at least one writer at least one further object within said model, and processing said further object.
12. A method according to claim 10 or 11, further comprising:
 - identifying a further writer configured to identify an entity within said hierarchical data structure to which data is to be written.
13. A method according to claim 12, wherein said identifying an entity comprises:
 - attempting to locate an entity within said hierarchical data structure to which data should be written; and
 - if said attempt is unsuccessful, creating an appropriate entity.

14. A method according to any one of claims 9 to 13, wherein each writer is a writer object which is an instance of a respective Java writer class.
15. A method according to claim 14, wherein each writer class has a corresponding writer factory class.
16. A method according to claim 15, further comprising:
 - registering each writer factory class with a writer lookup object;
 - providing details of the or each object to be processed to said writer lookup object; and
 - identifying one or more factory classes which should be used to create writer objects.
17. A method of generating model data representing a model of a composite user interface comprising a plurality of user interface elements provided by at least one source application, the method comprising:
 - modelling at least part of a user interface provided by the or each source application; and
 - modelling relationships between the at least part of the user interface provided by the or each source application and the composite user interface.
18. A method according to claim 17, wherein the model is adapted for use in generating a composite application.
19. A method according to claim 17 or 18, wherein modelling at least part of the user interface provided by the or each source application comprises:
 - defining a plurality of source flow items each comprising a specified source user interface page provided by a source application; and
 - defining relationships between said plurality of source flow items.
20. A method according to claim 19, wherein modelling at least part of the user interface provided by the or each source application further comprises:

defining at least one page element within each specified source user interface page.

21. A method according to claim 19 or 20, wherein modelling at least part of the user interface provided by the or each source application further comprises:

defining at least one flow control condition;

associating a flow control condition with at least one of said plurality of source flow items.

22. A method according to claim 19, 20, or 21, wherein modelling at least part of the user interface provided by the or each source application further comprises:

defining request parameters used to obtain each specified source user interface page.

23. A method according to claim 19, 20, 21 or 22, wherein modelling at least part of the user interface provided by the or each source application further comprises:

defining at least one rule for each specified source user interface page which can be applied to enable recognition of the associated specified source user interface page.

24. A method according to claim 23, wherein the or each rule is specified using a regular expression, or a path expression.

25. A method according to any one of claims 17 to 24, wherein modelling at least part of the user interface provided by the or each source application further comprises:

creating a plurality of objects which are instances of classes defined in an object oriented programming language.

26. A method according to any one of claims 17 to 25, wherein modelling relationships between the at least part of the user interface provided by the or each source application and the composite user interface comprises:

combining at least part of a plurality of source application models.

27. A method according to any one of claims 17 to 26, further comprising:

defining a plurality of composite flow items each comprising a specified user interface page; and

defining relationships between said plurality of composite flow items.

28. A method according to claim 27 as dependent upon claim 19, wherein at least one composite flow item is a source flow item, and said specified user interface page is a specified source user interface page.

29. A method according to claim 27 or 28, wherein at least one specified user interface page is a composite user interface page.

30. A method according to claim 29 as dependent upon claim 20, further comprising:

modelling manipulations which are applied to said at least one page element within a specified source user interface page to create said composite user interface page.

31. A method according to claim 30, further comprising:

specifying an ordered plurality of manipulations to be carried out to create said composite user interface page.

32. A method according to any one of claims 17 to 31, further comprising modelling at least one further user interface element to be included in the composite user interface.

33. A method according to any one of claims 17 to 32, further comprising:

processing said model to generate rules for communication between said composite user interface and said at least one source application.

34. A method according to claim 33, further comprising:

storing said rules within a hierarchical data structure comprising a plurality of entities.

35. A method according to claim 34, further comprising:

storing within said hierarchical data structure an entity representing the composite user interface; and

associating with said entity a data group providing configuration data for the composite user interface.

36. A method according to claim 35, further comprising:

storing within said hierarchical data structure details of a plurality of service entities representing processing modules which are together adapted to process user requests input to said composite user interface to produce one or more requests to at least one source application.

37. A method according to claim 36, wherein at least some of said service entities have an associated data group storing configuration data.

38. A method according to claim 37, wherein one of said service entities is an aggregation service entity representing an aggregation service configured to generate source application requests from a user request.

39. A method according to claim 38, wherein said aggregation service entity comprises:

a child entity representing the composite user interface; and
said child entity has at least one child entity representing a source application.

40. A method according to any one of claims 34 to 39, wherein said rules are generated using a plurality of writers each writer being associated with an entity in said hierarchical data structure, and being adapted to write data to a data group associated with the respective entity.

41. A method according to claim 40, wherein processing said model comprises:

selecting one or more objects within said model;
determining one or more writers to be invoked to write data from the or each object to data groups in said hierarchical data structure; and
invoking the or each writer to write data to said hierarchical data structure.

42. A method according to claim 41, further comprising:
determining from said at least one writer at least one further object within said model, and processing said further object.
43. A method according to claim 41 or 42, further comprising:
identifying a further writer configured to identify an entity within said hierarchical data structure to which data is to be written.
44. A method according to claim 43, wherein identifying an entity comprises:
attempting to locate an entity within said hierarchical data structure to which data should be written; and
if said attempt is unsuccessful, creating an appropriate entity.
45. A method according to any one of claims 40 to 44, wherein each writer is a writer object which is an instance of a respective writer class.
46. A method according to claim 45, wherein each writer class has a corresponding writer factory class.
47. A method according to claim 46, further comprising:
registering each writer factory class with a writer lookup object;
providing details of the or each object to be processed to said writer lookup object; and
identifying one or more factory classes which should be used to create writer objects.
48. A data carrier carrying computer program code means to cause a computer to carry out a method according to any preceding claim.
49. A computer apparatus comprising:
a program memory containing processor readable instructions; and

a processor for reading and executing the instructions contained in the program memory;

wherein said processor readable instructions comprise instructions controlling the processor to carry out the method of any one of claims 1 to 47.

50. Apparatus adapted to generate model data representing a model of a composite user interface comprising a plurality of user interface elements provided by at least one source application, the method comprising:

first generating means for generating model data representing a model of at least part of a user interface provided by the or each source application; and

second generating means for generating model data representing a model of relationships between the at least part of the user interface provided the or each source application and the composite user interface.

51. Apparatus according to claim 50, wherein said first generating means comprises:

first defining means adapted to define a plurality of source flow items each comprising a specified source user interface page provided by a source application; and

second defining means adapted to define relationships between said plurality of source flow items.

52. Apparatus according to claim 51, wherein said first generating means comprises means for defining at least one page element within each specified source user interface page.

53. Apparatus according to claim 51 or 52, wherein said first generating means comprises means for defining at least one flow control condition, and means for associating a flow control condition with at least one of said plurality of source flows.

54. Apparatus according to claim 51, 52, or 53, wherein said first generating means comprises means for defining request parameters used to obtain each specified source user interface page.

55. Apparatus according to claim 51, 52, 53 or 54, wherein said first generating means comprises means for defining at least one rule for each specified source page which can be applied to enable recognition of the associated specified source page.

56. Apparatus according to claim 55, wherein said at least one rule is specified using a regular expression, or a path expression.

57. Apparatus according to any one of claims 50 to 56, wherein said first generating means comprises means for creating a plurality of objects which are instances of classes defined in an object oriented programming language.

58. Apparatus according to any one of claims 50 to 57, wherein said second generating means comprises combining means for combining at least part of a plurality of source application models.

59. Apparatus according to any one of claims 50 to 58, further comprising defining means for defining a plurality of composite flow items each comprising a specified user interface page, and defining means for defining relationships between said plurality of composite flow items.

60. Apparatus according to claim 59 as dependent upon claim 51, wherein at least one composite flow item is a source flow item, and said specified user interface page is a specified source user interface page.

61. Apparatus according to claim 59 or 60, wherein at least one specified user interface page is a composite user interface page.

62. Apparatus according to claim 61 as dependent upon claim 52, further comprising means for modelling manipulations which are applied to said at least one page element within a specified source interface page to create said composite user interface page.

63. Apparatus according to claim 62, further comprising means for specifying an ordered plurality of manipulations to be carried out to create said composite user interface page.

64. Apparatus according to any one of claims 50 to 63, further comprising modelling at least one further user interface element to be included in the composite user interface.

65. Apparatus according to any one of claims 50 to 64, further comprising processing means adapted to process said model to generate rules for communication between said composite user interface and said at least one source application.

66. Apparatus according to claim 65, further comprising storage means for storing said rules, said storage means being adapted to store said rules in a hierarchical data structure comprising a plurality of entities.

67. Apparatus according to claim 66, wherein said storage means is adapted to store in said hierarchical data structure an entity representing the composite user interface and to store a data group providing configuration data for the composite user interface

68. Apparatus according to claim 67, wherein said storage means is adapted to store within said hierarchical data structure details of a plurality of service entities representing processing modules which are together adapted to process user requests input to said composite user interface to produce one or more requests to at least one source application.

69. Apparatus according to claim 68, wherein at least some of said service entities have an associated data group storing configuration data.

70. Apparatus according to claim 69, wherein one of said service entities is an aggregation service entity representing an aggregation service configured to generate source application requests from a user request.

71. Apparatus according to claim 70, wherein said aggregation service entity comprises:

a child entity representing the composite user interface; and
said child entity has at least one child entity representing a source application.

72. Apparatus according to any one of claims 66 to 71, comprising a plurality of writers adapted to write said rules, each writer being associated with an entity in said hierarchical data structure, and being adapted to write data to a data group associated with the respective entity.

73. Apparatus according to claim 72, wherein said processing means comprises:
selecting means for selecting one or more objects within said model;
determining means for determining one or more writers to be invoked to write data from the or each object to data groups in said hierarchical data structure; and
invoking means for invoking the or each writer to write data to said hierarchical data structure.

74. Apparatus according to claim 73, further comprising:
determining means for determining from said at least one writer at least one further object within said model; and
processing means for processing said further object.

75. Apparatus according to claim 73 or 74, further comprising identifying means for identifying a further writer configured to identify an entity within said hierarchical data structure to which data is to be written.

76. Apparatus according to claim 75, wherein said identifying means comprises:
locating means adapted to attempt to locate an entity within said hierarchical data structure to which data should be written;
analysis means to determine whether said attempt is successful or unsuccessful;
and

creating means for creating an appropriate entity if said analysis means determines that said attempt is unsuccessful.

77. Apparatus according to any one of claims 72 to 76, wherein each writer is a writer object which is an instance of a respective writer class.

78. Apparatus according to claim 77, further comprising a writer factory class associated with each writer class.

79. Apparatus according to claim 78, further comprising:

registration means for registering each writer factory class with a writer lookup object;

means for providing details of the or each object to be processed to said writer lookup object; and

identifying means for identifying one or more factory classes which should be used to create writer objects.

80. A server configured to provide at least one composite user interface to at least one source application, said composite user interface comprising a plurality of user interface elements provided by said at least one source application, the server comprising:

a processor adapted to process a model representing said composite user interface to generate rules for communication between said composite user interface and said at least one source application.

81. A server according to claim 80, wherein said model comprises a model of at least part of a user interface provided by the or each source application and a model of relationships between the at least part of the user interface provided the or each source application and the composite user interface.

82. A server according to claim 80 or 81, further comprising storage means storing a hierarchical data structure comprising a plurality of entities, said hierarchical data structure storing said rules.

83. A server according to claim 82, wherein said hierarchical data structure comprises an entity representing the composite user interface, and said entity is associated with a data group providing configuration data for the composite user interface.

84. A server according to claim 83, wherein said hierarchical data structure comprises a plurality of service entities representing processing modules which are together adapted to process user requests input to said composite user interface to produce one or more requests to at least one source application.

85. A server according to claim 84, wherein at least some of said service entities have an associated data group storing configuration data.

86. A server according to claim 85, wherein one of said service entities is an aggregation service entity representing an aggregation service configured to generate source application requests from a user request.

87. A server according to claim 86, wherein said aggregation service entity comprises:

- a child entity representing the composite user interface; and
- said child entity has at least one child entity representing a source application.

88. A server according to any one of claims 82 to 87, further comprising a plurality of writers adapted to generate said rules, each writer being associated with an entity in said hierarchical data structure, and being adapted to write data to a data group associated with the respective entity.

89. A server according to claim 88, wherein said processor comprises:

- selecting means for selecting one or more objects within said model;
- determining means for determining one or more writers to be invoked to write data from the or each object to said hierarchical data structure; and

means for invoking the or each writer to write data to said hierarchical data structure.

90. A server according to claim 89, further comprising means for determining using said at least one writer at least one further object within said model, and processing said further object.

91. A server according to claim 89 or 90, further comprising a further writer comprising identifying means configured to identify an entity within said hierarchical data structure to which data is to be written.

92. A server according to claim 91, wherein said identifying means comprises:
means for attempting to locate an entity within said hierarchical data structure to which data should be written; and
means for creating an appropriate entity if said attempt is unsuccessful.

93. A server according to any one of claims 88 to 92, wherein each writer is a writer object which is an instance of a respective writer class.

94. A server according to claim 93, wherein each writer class has a corresponding writer factory class.

95. A server according to claim 94, further comprising:
a writer lookup object, said writer lookup object comprising means adapted to register each writer factory class with said writer lookup object,
means for providing details of the or each object to be processed to said writer lookup object; and
means for identifying one or more factory classes which should be used to create writer objects.

96. A computer apparatus for generating a composite user interface for communication with a plurality of source applications, the apparatus comprising:

modelling means adapted to generate model data representing a model of said composite user interface in response to user input;

storage means for storing said model;

generating means for reading said model from said storage means, and generating a configuration data structure;

receiving means adapted to receive a request from a composite user interface;

generating means for generating a source application request to at least one of said source application in response to said request, in accordance with data stored in said configuration data structure; and

transmitting means for transmitting said source application request to said at least one of said source applications.

97. A method for modelling and generating a composite user interface comprising user interface elements provided by at least one source application comprising:

generating a source application model for each of the at least one source applications;

generating a composite application model using the or each source application model; and

processing said composite application model to generate rules for communication between said composite application and the or each source application.

98. A method for providing a composite user interface comprising a plurality of user interface elements provided by at least one source application, the method comprising:

monitoring operation of the composite user interface to obtain management data.

99. A method according to claim 98, further comprising modifying said composite user interface in response to said management data.

100. A method according to claim 99, wherein said modifying comprises deleting some of said plurality of user interface elements from said composite user interface.

101. A method according to claim 98, 99 or 100, further comprising producing data representing usage patterns of said composite user interface using said management data.

102. A method according to any one of claims 98 to 101, further comprising:
receiving at least one request message generated by said composite user interface;
producing at least one further message in response to said request message; and
forwarding said at least one further message to one of said at least one source applications.

103. A data carrier carrying computer program code means to cause a computer to carry out a method according to any one of claims 98 to 102.

104. A computer apparatus comprising:
a program memory containing processor readable instructions; and
a processor for reading and executing the instructions contained in the program memory;
wherein said processor readable instructions comprise instructions controlling the processor to carry out the method of any one of claims 98 to 102.

105. Apparatus for providing a composite user interface comprising a plurality of user interface elements provided by at least one source application, the apparatus comprising:
monitoring means for monitoring operation of the composite user interface to obtain management data.

106. Apparatus according to claim 105, further comprising modifying means adapted to modify said composite user interface in response to said management data.

107. Apparatus according to claim 106, wherein said modifying means is adapted to delete some of said plurality of user interface elements from said composite user interface.

108. Apparatus according to claim 105, 106 or 107, further comprising means for producing data representing usage patterns of said composite user interface using said management data.

109. Apparatus according to any one of claims 105 to 108, further comprising:
receiving means for receiving at least one request message generated by said composite user interface;
means for producing at least one further message in response to said request message; and
means for forwarding said at least one further message to one of said at least one source applications.

110. A method for generating a composite user interface comprising a plurality of user interface elements provided by at least one source application, the method comprising selecting said composite user interface from a plurality of predefined composite user interfaces on the basis of at least one predefined parameter.

111. A method according to claim 110, wherein said at least one predefined parameter comprises a parameter relating to at least one of time of day and date.

112. A method according to claim 110 or 111, wherein said at least one predefined parameter comprises a parameter relating to usage statistics of said composite user interface.

113. A method according to claim 110, 111 or 112, wherein said at least one predefined parameter comprises a parameter relating to a marketing campaign operated by a business enterprise using said composite user interface.

114. A method according to any one of claims 110 to 113, further comprising:
receiving at least one request message generated by said composite user interface;
producing at least one further message in response to said request message; and

forwarding said at least one further message to one of said at least one source applications.

115. A method according to claim 114, further comprising:

storing details of a plurality of user interface elements which are expected in response to said at least one further message;

storing data indicating that at least one of said expected user interface elements is mandatory, and at least one of said expected user interface elements is non-mandatory

receiving a plurality of user interface elements in response to said at least one further message;

generating a part of said composite user interface when all mandatory user interface elements have been received.

116. A method according to claim 115, wherein at least two of said plurality of predefined composite user interfaces comprise identical source user interface elements.

117. A method according to claim 116, wherein said at least two of said plurality of predefined composite user interfaces have different mandatory user interface elements.

118. A method according to any one of claims 110 to 117, wherein at least two of said plurality of predefined composite user interfaces comprise different source user interface elements.

119. Apparatus for generating a composite user interface comprising a plurality of user interface elements provided by at least one source application, the apparatus comprising:

storage means storing a plurality of predefined composite user interfaces; and

selecting means adapted to select said composite user interface from one of said plurality of predefined composite user interfaces on the basis of at least one predefined parameter.

120. Apparatus according to claim 119, wherein said at least one predefined parameter comprises a parameter relating to at least one of time of day and date.

121. Apparatus according to claim 119 or 120, wherein said at least one predefined parameter comprises a parameter relating to usage statistics of said composite user interface.

122. Apparatus according to claim 119, 120, or 121, wherein said at least one predefined parameter comprises a parameter relating to a marketing campaign operated by a business enterprise using said composite user interface.

123. Apparatus according to any one of claims 119 to 122, further comprising:
receiving means adapted to receive at least one request message generated by said composite interface;

generating means for producing at least one further message in response to said request message; and

transmitting means for forwarding said at least one further message to one of said at least one source application.

124. Apparatus according to claim 123, further comprising:

storage means adapted to store details of a plurality of user interface elements which are expected in response to said at least one further message;

storage means adapted to store data indicating that at least one of said expected user interface elements is mandatory, and at least one of said expected user interface elements is non-mandatory;

receiving means adapted to receive a plurality of user interface elements in response to said at least one further message;

generating means for generating a part of said composite user interface when all mandatory user interface elements have been received.

125. Apparatus according to claim 124, wherein at least two of said plurality of predefined composite user interface comprise identical source user interface elements.

126. Apparatus according to claim 125, wherein said at least two of said plurality of predefined composite user interfaces have different mandatory user interface elements.

127. Apparatus according to any one of claims 119 to 126, wherein at least two of said plurality of predefined composite user interface comprise different source user interface elements.

128. A data carrier carrying computer program code means to cause a computer to carry out a method according to any one of claims 110 to 118.

129. A computer apparatus comprising:

a program memory containing processor readable instructions; and

a processor for reading and executing the instructions contained in the program memory;

wherein said processor readable instructions comprise instructions controlling the processor to carry out the method of any one of claims 110 to 118.